SANDIA REPORT

SAND2005-6520
Unlimited Release
Printed June 2005

Analysis and Algorithms for Using Markov Processes in Systems Studies

Paul T. Boggs
Computational Science and Mathematics Research Department
Sandia National Laboratories
Livermore, CA 94551

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/ordering.htm

SAND2005-6520xx
Unlimited Release
Printed June 2005

Analysis and Algorithms for Using Markov Processes in Systems Studies

Paul T. Boggs
Computational Science and Mathematics Research Department
Sandia National Laboratories
Livermore, CA 94550
ptboggs@sandia.gov

Abstract

A Markov process model has been used for the DART systems analysis study. The basic design through analysis process is not immediately describable as a Markov process, but we show how a true Markov process can be derived and analyzed. We also show how sensitivities of the model with respect to the input values can be computed efficiently. This is useful in understanding how the results of this model can be used to determine strategies for investment that will improve the design through analysis process.

Acknowledgments

It is a pleasure to thank several of my colleagues for their help in carrying out this work. First, I would like to thank Mike Hardwick and Robert Clay for their discussions that led to the creation of the DART model. They were instrumental in creating a reasonably simple, but sufficient model that formed the basis for the creation of the Markov process. Next, I would like to thank Edward Walsh for his help in creating the overall package in which my software resides. Edward helped considerably in debugging my code and in getting it properly connected to his classes. I would also like to thank Vicki Howle for several useful conversations about the tree representation that led to the general algorithm of section 4 and Kevin Long and Steve Margolis for some general discussions about stochastic processes. Finally, I would like to thank Richard Byrd (on sabbatical leave from the University of Colorado) for his help in computing the sensitivities described in section 5.

Contents

Figures

Analysis and Algorithms for Using Markov Processes in Systems Studies

1 Introduction

Markov processes arise in many applications and are described in numerous books and papers on stochastic processes and operations research (see e.g., [3]. Basically, a Markov process consists of a finite number of states. If the process is in a given state, then at the completion of the work associated with that state the process transitions to another state. The particular state to which it transfers is determined by a fixed "transition matrix," say $\mathcal{T}$, where the element $\mathcal{T}_{i,j}$ gives the probability of transferring from state i to state j.

A particularly useful special case is a Markov process that has one or more "absorbing states." These are defined to be states that, once entered, are never left, i.e., if state i is an absorbing state then $\mathcal{T}_{i,j} = 0$ for $i \neq j$ and $\mathcal{T}_{i,i} = 1$. Well known methods exist that can compute the expected number of visits to each state before an absorbing state is entered. This model can be very useful in analyzing processes to determine which states are entered most often, how much time is spent in each state, and what the effects of investments in the process that change the transition probabilities would be. For example, if a state that is entered late in the process has a significant probability of returning to a very early state, then investing in changes that reduce the likelihood of such a transition may have a profound effect on the overall expected time through the process. Of course, the Markov process analysis cannot tell a manager exactly where to make such an investment since each process is different in that what causes the changes the the transition probabilities is not apparent in the model.

The project that motivated this study is the Sandia Design through Analysis Realization Team (DART) analysis activity that is attempting to improve this process (see Hardwick, *et al.* [2]). It was observed that the process could be described in ten steps and that, at the conclusion of each step, there was a probability of going to the next step as well as the probability of going back one or more steps. Of course, not all D–through–A projects used all ten steps, i.e., steps may be skipped, and that in going back to a previously entered state the transition probabilities would most likely change. The idea is that the first pass through a state, say the meshing part, might take two weeks and have a fairly high probability of going back to a design state. If the meshing state is entered a second time, it is most likely because a modification to only part of the mesh is required and, on average, say, one week will be spent and the transition probabilities to go forward rather than back would be greater. Since the probabilities change, this process in not a Markov process.

We need to be clear about the manner in which the times and the transition probabilities in the DART model change. We mean that for the first pass through any state, there is an

expected time and a set of transition probabilities. On the second pass, there is a (possibly) different expected time and set of transition probabilities. We allow this to continue, but only for a fixed number of passes, say p. In the p^{th} *and all subsequent* passes through any state, there is an expected time and set of transition probabilities that remain fixed thereafter. The DART analysis team did not believe that there was sufficient data to warrant a model where p was greater than 2. Thus the DART model has only two sets of expected times and two sets of transition probabilities.

In this paper, we discuss a strategy for creating a true Markov process from the model described above. First we provide (in section 2) the essential background on Markov processes that provides the basis for our analysis. Then, in section 3, we show how to create a Markov process from the DART model in a simple case. This is included for completeness since this was first considered to be sufficient for the analysis. In section 4, we show how to create the Markov process for a much more general case, which, as it turns out, is more efficient even for the special case of section 3. In section 5 we turn to the question of sensitivities and show how to compute the sensitivity of the total expected time with respect to both the transition probabilities and the expected times for each state. The case of sensitivities with respect to transition probabilities has some delicate features that require special consideration. Finally, in section 6, we discuss the implementation of these procedures in Java so that they are easily used in a code that creates a GUI for the analysts to use.

2 The Basics of Markov Processes

To make this report more self-contained, we include in this section a brief review of the basic results on Markov processes that are needed for the analyses that were used in [2]. This also serves to set the notation for subsequent sections.

As noted in the introduction, a Markov process consists of a finite number of states and a transition matrix that gives the probability of going from one state to another. To be specific, let the states be denoted by

$$s_i, \qquad i = 1, \ldots, N,$$

where N is the number of states in the system. Let $\mathcal{T} \in \mathcal{R}^{N \times N}$ be the transition matrix and recall that $\mathcal{T}_{i,j}$ is the transition probability from s_i to s_j. Since probabilities must be nonnegative and some transition must be made in s_i, it follows that

$$\sum_{j=1}^{N} \mathcal{T}_{i,j} = 1 \qquad \text{for } i = 1, \ldots, N.$$

That is, all of the row sums of $\mathcal{T}$ must be one. Such a matrix is called a "stochastic matrix."

If we let $y_0 \in \mathcal{R}^N$ be a vector with a one in the first position and zero in the other positions, we can interpret it as the initial state of the process, i.e., there is a 100% chance that the process is in s_1 at the beginning. It follows that for

$$y_1 = \mathcal{T}^{\mathsf{T}} y_0,$$

where superscript T means "transpose," the k^{th} element of y_1 is the probability of being in s_k after one step of the process. In general, for

$$y_p = \mathcal{T}^{\mathsf{T}} y_{p-1} = (\mathcal{T}^{\mathsf{T}})^p y_0,$$

it follows that the k^{th} element of y_p is the probability of being in s_k after p steps of the process.

Recall that a Markov process has an absorbing state, say s_a, if $\mathcal{T}_{a,a} = 1$. If a process is a Markov process with one or more absorbing states, we can easily reorder the states so that all of the absorbing states are at the end of the list of states. We can then partition $\mathcal{T}$ as follows:

$$\mathcal{T} = \begin{bmatrix} S & R \\ 0 & I \end{bmatrix},$$

where S are the transitions among the nonabsorbing states, R are the transitions from the nonabsorbing states to the absorbing states, and I is the identity matrix.

As above, let $w_0 \in \mathcal{R}^{N-N_a}$, where N_a is the number of absorbing states and let w_0 have a one in the first position and zeros elsewhere. This can be interpreted again as starting in s_1. We are interested in the number of times that each state will be visited before the process enters an absorbing state. Let

$$w_p = (S^{\mathsf{T}})^p w_0.$$

Since the k^{th} element of w_p is the probability of being in s_k after p steps of the process, we interpret the k^{th} element of v_p, where

$$v_p = \sum_{i=0}^{p} w_i, \tag{2.1}$$

as the expected number of visits to s_k given that we started in s_1. (Note that the summation starts at 0 so that the expected number of visits to s_1 is at least one. We can rewrite (2.1) as

$$v_p = \left[I + (S^{\mathsf{T}}) + (S^{\mathsf{T}})^2 + \cdots + (S^{\mathsf{T}})^p \right] w_0.$$

It can be proved that

$$\lim_{p\to\infty}\left[I+(S^{\mathsf{T}})+(S^{\mathsf{T}})^2+\cdots+(S^{\mathsf{T}})^p\right]=(I-S^{\mathsf{T}})^{-1}.$$

(This is only true under certain conditions, e.g., that all of the eigenvalues of S are less than 1, which they are in this case.) Let

$$F=(I-S^{\mathsf{T}}).$$

Then the expected number of visits to each state given that we start in s_1 is given by the elements of the vector

$$v\equiv v_\infty=F^{-1}w_0.$$

Finally, if the time that it takes to complete the work in s_i is denoted by t_i then it follows that the expected time for the process to enter an absorbing state is given by

$$T=v^{\mathsf{T}}t, \tag{2.2}$$

where t is the vector whose i^{th} element is t_i. This is the quantity that is of interest in the DART application.

3 Creating a Markov Process from the DART Model

To understand the procedure for creating a Markov process from the DART model, we first illustrate with a simple example. Assume that the DART process consists of three states and that we append a fourth state, the "done" state, that is an absorbing state. Assume also, as in section 1, that we only have two sets of times and transition probabilities associated with each state. For a standard Markov process, we can illustrate the process as in figure 1. There are fixed transition probabilities associated with each arrow in the figure. In the DART model, however, the probabilities change after the first pass, so we must modify this figure. Our procedure is based on replicating the states to take into account the changing probabilities. Note that in figure 1 each state has a probability of going forward only one step and a probability of going back to each preceding state. This is, in fact, what was assumed for the first DART model. In figure 2 we show how to create a Markov process that handles this case. In figure 2 the first row corresponds to the first pass data; all subsequent rows correspond to the second pass data. Note that the transition to the first occurrence of s_3 can occur from either the first pass through s_2 in the first row, or the second pass through s_2 in the second row. After that, any transition to s_3 must be to a second pass version in rows

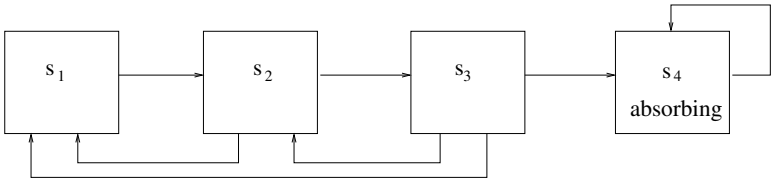

Figure 1. A standard Markov process allowing only one forward transition, but arbitrary backward transitions. If the transition probabilities change on subsequent passes through a state, this will no longer be a valid Markov process.

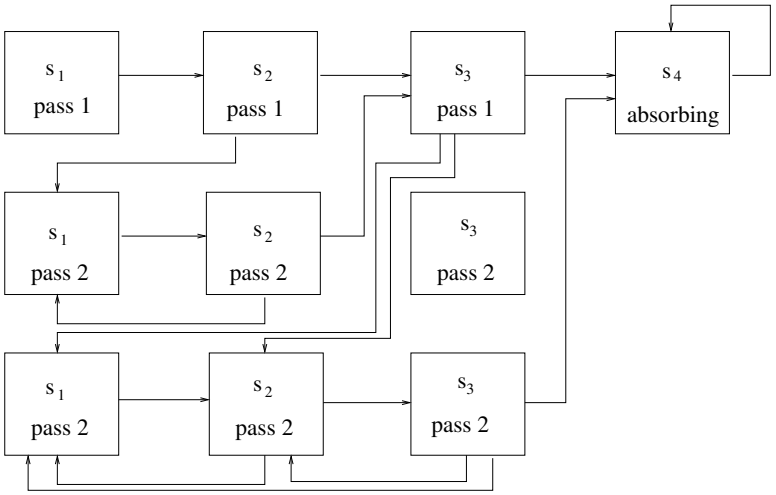

Figure 2. A Markov process derived from a model where transition probabilities change. Note that the nonabsorbing states are replicated to achieve a larger, but valid Markov process.

$2, 3, \ldots$. One can now observe that, at worst, we will need to add two rows, so there will be 9 nonabsorbing states and one absorbing state. A careful consideration shows that this will not work if a state is skipped on the first pass and that it is hard to generalize if there are more than two passes. Note also that not all of the replicated states are actually needed, e.g., state s_3 in the second row, but they are included for simplicity of the construction of the Markov process.

If the original process has N states and a "done" state is added, we will have to add N rows giving (N^2+1) states in the final Markov process. Since the DART model has only 10 states, we will have to solve linear systems of size 100, which is simple to compute. As noted above, there are replicated states that are unnecessary and we could remove them. Doing so, however, is more work than just ignoring them. The software for solving the system of equations is explained later.

$$\begin{bmatrix} 0 & x & 0 & x \\ x & 0 & x & 0 \\ 0 & x & 0 & x \\ 0 & 0 & 0 & x \end{bmatrix}$$

Figure 3. Transition matrix for first pass

$$\begin{bmatrix} 0 & x & x & x \\ x & 0 & 0 & x \\ 0 & x & 0 & x \\ 0 & 0 & 0 & x \end{bmatrix}$$

Figure 4. Transition matrix for all later passes

The first version of the DART tool was used by a set of analysts. At least two of these analysts violated our assumption by either skipping states or by having transitions to more than one forward state. This is in line with the experience of many modelers and programmers that users often find ways to use the models or codes that developers never considered. On the positive side, however, this led to the development of a better and more general strategy that is described next.

4 Creating a Markov Process from a General Model

As noted in the previous section, we were able to replicate states to produce a Markov process from a model where the transition probabilities changed on the second and subsequent passes through a state. In this section, we develop an algorithm to handle a more general case than that of section 3. Specifically, we allow arbitrary transitions to any state and an arbitrary, but finite, number of passes through a state before the transitions become fixed. Recall that in section 3, we allowed forward transitions to be only to the next state with no skipping of states allowed.

The method for handling this case is derived from the recognition that transition through a Markov process can be represented by a tree and that, associated with each new, or replicated, state, we need to know how many times each of the other states has been visited. The idea for constructing the states of a Markov process is best illustrated by an example: Consider a process with four states, where s_4 is the absorbing state. Suppose the transition probabilities for the first pass through any state are given by the transition matrix in figure 3 and the transition probabilities for the second and subsequent passes are given by the transition matrix in figure 4. Note that the actual probabilities don't matter, so we have just put an x in the matrix to represent a nonzero probability.

Since we need to keep track of the number of times each state has been entered, we devised a notation to name the states of the Markov process with characters that contains these counts. In the case of our example, we use a sequence of six integers for creating and

naming the states. For example, if a state is represented by

$$[2-0-1-0-1-0],$$

it has the following interpretation:

2	The number of the state in the original model
0	The number of times this state has been entered
1	The number of times state 1 has been entered
0	The number of times state 2 has been entered
1	The number of times state 3 has been entered
0	The number of times state 4 has been entered.

Since, in this example, there are only two distinct transition matrices, we only need to know if we are in the first pass or in the subsequent passes. Thus, for our purposes, the number of times a state is entered needs to be only 0 or 1.

Now we can can begin the process of generating a list all of the states that are possible in the final Markov process. The first state in the list will be s_1 in the first pass with all other states having been entered zero times. This state is represented by

$$[1-0-0-0-0-0]$$

and the list contains just this state. We see from the transition matrix in figure 3 that we can transition from s_1 to s_2 or s_4. Thus we add the following states to the list:

$$[2-0-1-0-0-0]$$
$$[4-0-1-0-0-0].$$

Note that since we have now passed through s_1, there is a 1 in the position corresponding to the number of times s_1 has been entered. We now have to process each new state on the list, adding states to the list if they do not already appear. For the second state, we see that we can transition to states s_1 or s_3. Thus the states that we add are

$$[1-1-1-1-0-0]$$
$$[3-0-1-1-0-0].$$

Observe that the first of these has a 1 in the second position, indicating that this is the second pass through s_1. Also, there is a 1 in the fourth position since s_2 has now been entered. The absorbing state, s_4, is a special case and only has to be in the list once regardless of the number of passes through any other state since, no transition out of s_4 is allowed.

Continuing, we now add the states that can be reached from the two recently added states. From figure 4 we see that from s_1 in the second pass, we can transition to s_2, s_3, or s_4. We must take into account states already visited, so we consider that states

$$[2-1-1-1-0-0]$$
$$[3-0-1-1-0-0]$$
$$[4-0-1-1-0-0].$$

But now we see that the state [3 - 0 - 1 - 1 - 0 -0] is already in the list, so we don't add it again. We still have [3 - 0 - 1 - 1 - 0 -0] on the list and have not added the states that it can transition to. These are states s_2 in its second pass and the absorbing state. As above, we only need to add

$$[2-1-1-1-1-0],$$

which is a new state. The only new state from here is

$$[1-1-1-1-1-0],$$

which we add. This state then generates the final state to be added,

$$[3-1-1-1-1-0].$$

Thus, the final list of states for our Markov process is

$$[1-0-0-0-0-0]$$
$$[2-0-1-0-0-0]$$
$$[1-1-1-1-0-0]$$
$$[3-0-1-1-0-0]$$
$$[4-0-1-0-0-0]$$
$$[2-1-1-1-0-0]$$
$$[2-1-1-1-1-0]$$
$$[1-1-1-1-1-0]$$
$$[3-1-1-1-1-0].$$

We now proceed to create the transition matrix, $\mathcal{T}$, for this list of states. The entries are easily determined from the list itself and from the transition probabilities in figures 3 and 4. From $\mathcal{T}$ we create F and solve for v, the expected number of visits to each state,

$$v = F^{-1}w_0, \tag{4.3}$$

where w_0 is a vector with a one in the first position and zeros elsewhere. Observe that this is correct since the first state in our Markov process corresponds to the first state in the original.

It is a simple matter to map v to the number of visits to each state in the original model. To do this, let $\sigma_i,\ i = 1,\ldots,N_{MP}$, where N_{MP} is the total number of states in the Markov process. Now, the expected number of visits to s_i in pass j is given by

$$\eta_{i,j} = \sum_{k\in\kappa} v_k \tag{4.4}$$

where $\kappa = \{k : \text{the first digit of } \sigma_k \text{ is } i \text{ and the second digit is } j\}$. Thus the total time in terms of the original problem is

$$T = \sum_{i=1}^{N}\sum_{j=1}^{2} \eta_{i,j}\tau_{i,j}, \tag{4.5}$$

where $\tau_{i,j}$ is the time spent in state i during pass j. (Here we have used a two-pass system to illustrate the approach.)

It is easy to see how this procedure can be generalized. There are already cases where the forward transition is not just to the next state, e.g., from s_1 we can go to s_2 or s_4. In addition, if there are k passes before the transition probabilities stay fixed instead of two as above, then the count for each state goes from 0 to $k-1$. This procedure also has another advantage over the algorithm in section 3, namely that it only creates states that can be reached instead of creating the full N^2 states. Finally, the algorithm for creating the list of states is easy to code, due to the tree-like structure of the data.

5 Sensitivities

As noted in the introduction, it is of interest to use this model to gain a more complete understanding of the process of design through analysis and to guide the investment of funds to points in the process where the greatest reduction in time can be achieved. In this section, we show how to compute the sensitivity (or derivative) of the total time with respect to the the times associated with each state and, more importantly, with respect to the transition probabilities.

The derivatives of the time with respect to the times spent in the original s_i can be computed as follows from (4.5):

$$\frac{\partial T}{\partial \tau_{i,j}} = \eta_{i,j}.$$

This says that the sensitivity or derivative with respect to the time spent in s_i is simply the expected number of visits to s_i in pass j.

The derivative with respect to a transition probability is more difficult to compute and interpret. To derive this, we first show how to compute the derivative of the inverse of a matrix with respect to a perturbation in the matrix. In particular, let $A \in \mathcal{R}^{N\times N}$ be a given matrix and let $P \in \mathcal{R}^{N\times N}$ be a perturbation. We will discuss restrictions on P later. We define

$$\partial_P A^{-1} = \lim_{\varepsilon\to 0} \frac{(A+\varepsilon P)^{-1} - A^{-1}}{\varepsilon}.$$

We can write

$$\begin{aligned}(A+\varepsilon P)^{-1} - A^{-1} &= (A+\varepsilon P)^{-1}\,[A-(A+\varepsilon P)]A^{-1} \\ &= (A+\varepsilon P)^{-1}\,[-\varepsilon P]A^{-1}.\end{aligned}$$

Thus

$$\begin{aligned}\lim_{\varepsilon\to 0} \frac{(A+\varepsilon P)^{-1} - A^{-1}}{\varepsilon} &= \lim_{\varepsilon\to 0} \frac{(A+\varepsilon P)^{-1}(-\varepsilon P)A^{-1}}{\varepsilon} \\ &= -A^{-1}PA^{-1}.\end{aligned}$$

For technical reasons, we need to normalize the perturbation and the appropriate normalization is the so-called Frobenius norm of P, which is defined to be the square root of the sum of the squares of of all of the elements of P. This norm is denoted by $\|P\|_F$. Thus for a normalized perturbation, we have

$$\partial_P A^{-1} = -\frac{A^{-1}PA^{-1}}{\|P\|_F}. \tag{5.6}$$

Note that this quantity is an $(N\times N)$ matrix.

In our case, we have

$$A = F = (I - S^{\mathsf{T}}).$$

As above, we want to compute the derivative of the total time with respect to a perturbation in the transition probability. Thus, using 4.5 we can derive the following:

$$\frac{\partial T}{\partial P} = \frac{\tau^{\mathsf{T}} F^{-1} P F^{-1} w_0}{\|P\|_F}, \tag{5.7}$$

where w_0 is the vector with a 1 in the first position and zeros elsewhere.

Although (5.7) is theoretically valid for any perturbation, some restrictions are necessary in our case. In particular, changing just one transition probability will result in a transition matrix that is no longer valid in that the row sum will no longer be 1. Thus to maintain the row sums to be 1, the row sums of the perturbation P must be 0. It is not necessary to restrict the perturbations so that the probabilities remain between 0 and 1 since the normalization would obviate any such restrictions. The interpretation is that (5.7) gives the instantaneous change of the time in the "direction" of the perturbation P.

In using (5.7) we will want to consider several, and possibly many, perturbations to help in understanding and using the results of the model. It is interesting to note that the calculation of these sensitivities can be done very efficiently. First, note that we have already computed

$$\eta = F^{-1} w_0$$

and that we can compute, once and for all,

$$q = \left(F^{\mathsf{T}}\right)^{-1} \tau. \tag{5.8}$$

Thus, for any perturbation P we only need to compute

$$q^{\mathsf{T}} P \eta.$$

It is not immediately obvious what perturbations to consider. After discussions with the members of the DART analysis team, we decided that we should compute perturbations with respect to the transitions corresponding to each original state separately. Since any original state may be replicated several times in the derived Markov process, we create a perturbation matrix P with perturbations in rows corresponding to that state. In particular, we calculate the number of forward transitions and the number of backward transitions, n_f and n_b respectively, and we put a $1/n_f$ in any column corresponding to a transition to a forward state and a $-1/n_b$ in any column corresponding to a transition to a backward state. The rationale is that we are interested in changes that increase the probability of going on and, correspondingly, decrease the chances of going back. Note that it is not clear *a priori* what actions will create these changes.

6 Software

The software is described at the high level in [2]. Here we briefly describe the two Java classes we wrote to do the basic analysis and the sensitivities.

The code is embedded in a package that collects the following data from an analyst. Recall that the analysis is assumed to start in the first state and that there is a time required and a set of transition probabilities for the first pass through the states and a different time and set of transition probabilities for each subsequent pass. First, the analyst gives amount of time spent in each state for the first pass through that state. Then he/she supplies the percent of time required for subsequent passes through the state. Next, the analyst supplies the two sets of transition probabilities. This data is made available to the analysis and sensitivities classes.

The first class, Analysis2, creates Markov process from the data collected as described above. The list of states required for the Markov process is built and then the transition matrix is constructed as described in section 4. Next, the matrix F is constructed and factored using the linear algebra routines in JLApack, [1]. Finally the system (4.3) is solved using these factors.

The second class, Sensitivities, computes the sensitivities as described in section 5. The system (5.8) is solved, also using the factorization of F. Then the perturbations are computed and the sensitivities calculated.

References

[1] J. Dongarra. Jlapack download page. http://icl.cs.utk.edu/projects/f2j/download.html.

[2] Michael Hardwick, Robert Clay, Paul Boggs, Edward Walsh, Alex Larzelere, and Alan Altshuler. Dart systems analysis. Technical report, Sandia National Laboratories, 2005.

[3] Edward P.C. Kao. *An Introduction to Stochastic Processes*. Duxbury Press, Belmont, California, 1997.

DISTRIBUTION:

1 MS 9159
Mike Hardwick, 8964

1 MS 9159
Robert Clay, 8964

1 MS 9159
Edward Walsh, 8964

3 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616

1 MS 0612
Review & Approval Desk, 9612